

Value: 0020

MM MM AAAAAA RRRRRRRR 88888888 LL IIIII
MM MM AAAAAA RRRRRRRR 88888888 LL IIIII
MMMM MMMM AA AA RR RR BB BB LL IIIII
MMMM MMMM AA AA RR RR BB BB LL IIIII
MM MM MM AA AA RR RR BB BB LL IIIII
MM MM AA AA RR RR BB BB LL IIIII
MM MM AA AA RRRRRRRR 88888888 LL IIIII
MM MM AA AA RRRRRRRR 88888888 LL IIIII
MM MM AAAAAAAA RR RR BB BB LL IIIII
MM MM AAAAAAAA RR RR BB BB LL IIIII
MM MM AA AA RR RR BB BB LL IIIII
MM MM AA AA RR RR 88888888 LLLLLLLL IIIII
MM MM AA AA RR RR 88888888 LLLLLLLL IIIII
LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLL LLLL SSSSSSSS
LLLLLLL LLLL SSSSSSSS

1 0001 0 MODULE MARBLI(IDENT='V04-000',MAIN=CONVERT XTITLE'MARS to BLISS Macro Converter')=

2 0002 1 BEGIN

3 0003 1

4 0004 1 !*****

5 0005 1 *

6 0006 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY

7 0007 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

8 0008 1 * ALL RIGHTS RESERVED.

9 0009 1 *

10 0010 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED

11 0011 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE

12 0012 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER

13 0013 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY

14 0014 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY

15 0015 1 * TRANSFERRED.

16 0016 1 *

17 0017 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE

18 0018 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT

19 0019 1 * CORPORATION.

20 0020 1 *

21 0021 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS

22 0022 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

23 0023 1 *

24 0024 1 *

25 0025 1 !*****

26 0026 1

27 0027 1 ++

28 0028 1

29 0029 1 ABSTRACT:

30 0030 1 The routines in this module, along with XPORT

31 0031 1 convert a specifically formatted MARS macro file into a

32 0032 1 BLISS macro file.

33 0033 1

34 0034 1 The logic in this module is somewhat ad hoc and

35 0035 1 any perturbation of the format of the MARS file may result

36 0036 1 in these routines not working.

37 0037 1

38 0038 1 AUTHOR:

39 0039 1 P.C. Marks, CREATION DATE: 3 FEB 77

40 0040 1

41 0041 1 MODIFIED BY:

42 0042 1

43 0043 1 V03-001 MLJ0093 Martin L. Jack, 14-Jul-1982 11:38

44 0044 1 If the ".MACRO" line contains the string "CJFS", change the

45 0045 1 routine prefix from SYSS to CJFS. Do the same for RUF.

46 0046 1

47 0047 1 V02-003 BLS0073 Benn Schreiber 26-Aug-1981

48 0048 1 Disable per-macro output

49 0049 1

50 0050 1 V02-002 APL0001 Al Lehotsky 4-Jun-1981

51 0051 1 Change for 31-character names which was missed earlier.

52 0052 1

53 0053 1 V02-001 BLS0055 Benn Schreiber 3-Jun-1981

54 0054 1 Use BLISS linkage and GENERAL addressing mode

55 0055 1

56 0056 1 A.P. Lehotsky 31-Oct-79 Transport to VAX. Suppress "()" in

57 0057 1 macros without arguments, e.g. \$HIBER

58	0058	1	;	
59	0059	1	!--	
60	0060	1		
61	0061	1		
62	0062	1	FORWARD ROUTINE	
63	0063	1	CONVERT	
64	0064	1	CONVMACRO:	NOVALUE.
65	0065	1	OUTLIST,	
66	0066	1	CONVARGLIST:	NOVALUE:
67	0067	1		
68	0068	1	MACRO	

>Main routine
Convert a .MACRO definition
Output a parameter list
Convert argument list of a .MACRO

4F
53
41

```

69 0069 1 VERSTR= 'MARBLI V03-001' %;
70 0070 1 LIBRARY 'SYSSLIBRARY:XPORT';
71 0071 1
72 0072 1 LITERAL
73 0073 1 MAX_NO_ARGS= 63,                                | Max number of macro arguments
74 0074 1 MAX_ARG_LIST= 65*MAX_NO_ARGS,                 | Max characters in formal arg list
75 0075 1 MAX_PEC_SIZE= 130;                            | Max input and output record length
76 0076 1
77 0077 1 MACRO
78 M 0078 1 REPEAT=
79 0079 1 WHILE 1 DO %.
80 0080 1
81 0081 1 ! Construct length and pointer parameters
82 M 0082 1
83 M 0083 1 CH$LEN PTR[]=
84 0084 1 %CHARCOUNT(%REMAINING), CH$PTR(UPLIT(%REMAINING)) %.
85 0085 1
86 0086 1
87 0087 1 ! Return the index of string S in context C
88 0088 1
89 M 0089 1 CH$INDEX(CL,C,S) =
90 0090 1 CH$FIND_SUB(CL, C, %CHARCOUNT(S), CH$PTR(UPLIT(S))) %,
91 0091 1
92 0092 1 ! Return the size of the zero-truncated output buffer
93 0093 1
94 M 0094 1 TRUNCATED_OUTPUT =
95 M 0095 1 CR$DIFF(CH$FIND_CH(MAX_REC_SIZE+1, CH$PTR(OUTPUT_RECORD),0),
96 0096 1 CH$PTR(OUTPUT_RECORD)) %;
97 0097 1
98 0098 1
99 0099 1 OWN
100 0100 1 !+
101 0101 1 ! Declarations for processing a text line for macro declaration and
102 0102 1 the associated argument list.
103 0103 1 -
104 0104 1 INPUT_RECORD: VECTOR[CH$ALLOCATION(MAX_REC_SIZE)],
105 0105 1 INPUT_PTR,
106 0106 1 INPUT_LENGTH,
107 0107 1
108 0108 1 ! Guarantee at least 1 zero byte at end of output
109 0109 1 record s.t. TRUNCATED_OUTPUT macro cannot fail.
110 0110 1
111 0111 1 OUTPUT_RECORD: VECTOR[CH$ALLOCATION(MAX_REC_SIZE+1)],
112 0112 1 OUTPUT_PTR,
113 0113 1
114 0114 1 ARG_PTR,
115 0115 1 ARG_LENGTH,
116 0116 1 ARG_LIST: VECTOR[CH$ALLOCATION(MAX_ARG_LIST)],
117 0117 1
118 0118 1 CALL_PTR,
119 0119 1 CALL_LENGTH,
120 0120 1 CALL_LIST: VECTOR[CH$ALLOCATION(MAX_ARG_LIST)],
121 0121 1
122 0122 1 VAR_ARGS:
123 0123 1
124 0124 1 OWN
125 0125 1 !+

```

: 126 0126 1 I/O related declarations.
: 127 0127 1 !-
: 128 0128 1
: 129 0129 1 TERMINAL: \$XPO_I08(),
: 130 0130 1 INPUT: \$XPO_I08(),
: 131 0131 1 OUTPUT: \$XPO_I08();

```
133 0132 1 ROUTINE CONVERT =
134 0133 1 ++
135 0134 1 FUNCTIONAL DESCRIPTION:
136 0135 1 This is the main routine of this module.
137 0136 1 The chief function is to examine an input line and determine
138 0137 1 whether it should be ignored or processed by other routines.
139 0138 1
140 0139 1 IMPLICIT INPUTS:
141 0140 1
142 0141 1 OWN storage
143 0142 1
144 0143 1 IMPLICIT OUTPUTS:
145 0144 1
146 0145 1 OWN storage
147 0146 1
148 0147 1 ROUTINE VALUE:
149 0148 1
150 0149 1 Success or an XPORT completion code
151 0150 1 --
152 0151 2 BEGIN
153 0152 2 LOCAL
154 0153 2 PRINT_COMMENTS; ! Flag to pass comments and blank lines to output
155 0154 2
156 0155 2 $XPO_IOP_INIT(IOP=TERMINAL);
157 0156 2 $XPO_IOP_INIT(IOP=INPUT);
158 0157 2 $XPO_IOP_INIT(IOP=OUTPUT);
159 0158 2
160 0159 2 $XPO_OPEN( IOP=TERMINAL, FILE_SPEC=$XPO_INPUT);
161 0160 2 ! $XPO_PUT( IOP=TERMINAL,
162 0161 2 ! STRING=(%STRING('System-Service Macro Translator ',%EXPAND VERSTR)) );
163 0162 2
164 0163 2 REPEAT
165 0164 3 BEGIN
166 0165 4 IF NOT $XPO_GET(IOP=TERMINAL, PROMPT='Input file (STARLET.MAR): ' )
167 0166 3 THEN
168 0167 3 RETURN XPOS_NORMAL;
169 0168 3
170 P 0169 3
171 P 0170 3 IF $XPO_OPEN(IOP=INPUT,
172 P 0171 3 ! FILE SPEC=TERMINAL[IOP$T_STRING], ! User supplied name
173 P 0172 4 ! DEFAULT='STARLET.MAR', ! The standard input
174 P 0173 3 FAILURE=XPOSIO_FAILURE)
175 P 0174 3 THEN EXITLOOP ! Got good input.
176 P 0175 2 END;
177 P 0176 2
178 P 0177 3 IF NOT $XPO_GET(IOP=TERMINAL, PROMPT='Output file (*.B32): ' )
179 P 0178 2 THEN
180 P 0179 2 RETURN XPOS_NORMAL;
181 P 0180 2
182 P 0181 2 $XPO_OPEN(IOP=OUTPUT,
183 P 0182 2 ! FILE SPEC=TERMINAL[IOP$T_STRING],
184 P 0183 2 ! DEFAULT='*.B32',
185 P 0184 2 RELATED=INPUT[IOP$T_RESULTANT],
186 P 0185 2 OPTION=OUTPUT);
187 P 0186 2
188 P 0187 2 $XPO_PUT(IOP=OUTPUT,
```

```
189 0188 2      STRING=(%STRING('! Translated from MACRO-32 by ',%EXPAND VERSTR)) );  
190 0189 2      ! Set CSP pointer for the output record.  
191 0190 2  
192 0191 2      OUTPUT_PTR = CH$PTR(OUTPUT_RECORD);  
193 0192 2      PRINT_COMMENTS = 0;  
194 0193 2  
195 0194 2  
196 0195 2      ! Main loop:  
197 0196 2      Exit from this loop when an end of file is read  
198 0197 2      or an I/O error has occurred.  
199 0198 2  
200 0199 2      WHILE SXPO_GET(IOB=INPUT) DO  
201 0200 3      BEGIN  
202 0201 3      INPUT_LENGTH = .INPUT[IOB$H_STRING];  
203 0202 3      INPUT_PTR = .INPUT[IOB$A_STRING];  
204 0203 3  
205 0204 3      IF .INPUT_LENGTH LSS 0 THEN EXITLOOP;  
206 0205 3  
207 0206 3  
208 0207 3      IF .INPUT_LENGTH EQ 0  
209 0208 4      THEN  
210 0209 5      BEGIN  
211 0210 4      IF .PRINT_COMMENTS THEN SXPO_PUT(IOB=OUTPUT, STRING='')  
212 0211 3      END  
213 0212 4      ELSE  
214 0213 4      BEGIN  
215 0214 4      CH$COPY(.INPUT_LENGTH, .INPUT_PTR, 0, MAX_REC_SIZE, .OUTPUT_PTR);  
216 0215 4  
217 0216 4      INPUT_LENGTH = TRUNCATED_OUTPUT;  
218 0217 4  
219 0218 4      IF CH$RCHAR(.OUTPUT_PTR) EQ XC';'  
220 0219 5      THEN  
221 0220 5      BEGIN  
222 0221 5      Comment line  
223 0222 5      Note comments are printed only after line ";"* is read  
224 0223 5  
225 0224 5      CH$WCHAR(XC'!', .OUTPUT_PTR);  
226 0225 5      IF .PRINT_COMMENTS  
227 0226 5      THEN  
228 0227 5      SXPO_PUT(IOB=OUTPUT, STRING=(.INPUT_LENGTH, .OUTPUT_PTR));  
229 0228 5  
230 0229 5      IF CH$RCHAR(CH$PLUS(.OUTPUT_PTR, 1)) EQ XC'*'  
231 0230 5      THEN  
232 0231 5      PRINT_COMMENTS = 1  
233 0232 5  
234 0233 5  
235 0234 4      END  
236 0235 4      ELSE IF (CH$INDEX(.INPUT_LENGTH, .OUTPUT_PTR, '.MACRO')) NEQ 0  
237 0236 4      THEN  
238 0237 4      IF (CH$INDEX(.INPUT_LENGTH, .OUTPUT_PTR, '_S')) NEQ 0  
239 0238 4  
240 0239 4      Macro in "_S" form: Output it  
241 0240 4  
242 0241 4      CONVMACRO()  
243 0242 4  
244 0243 2  
245 0244 2      END  
END:
```

```

: 246
: 247
: 248
: 249
: 250
: 251
: 252
: 253
: 254
0245 2
0246 2
0247 2
0248 2
0249 2
0250 2
0251 2
0252 2
0253 1
0254 1

:     $XPO_CLOSE( IOB=INPUT);
:     $XPO_CLOSE( IOB=OUTPUT);
:     $XPO_PUT( IOB=TERMINAL, STRING='End MARBLI' );
:     $XPO_CLOSE( IOB=TERMINAL );
:     XPOS_NORMAL
:     END;

```

```

.TITLE MARBLI MARS to BLISS Macro Converter
.IDENT \V04-000\
```

```
.PSECT SPLIT$,NOWRT,NOEXE,2
```

```

41 54 53 28 20 65 54 55 50 4E 49 24 53 59 53 00000 P.AAA: .ASCII \SYSSINPUT\
20 3A 29 52 69 66 20 74 75 70 6E 49 00009 P.AAB: .ASCII \Input file (STARLET.MAR): \
52 41 4D 2E 54 45 4C 52 41 54 4C 52 00018
2E 2A 28 20 65 60 69 66 20 74 75 70 74 75 4F 00023 P.AAC: .ASCII \STARLET.MAR\
20 3A 29 32 33 42 2E 2A 00043 P.AAE: .ASCII \*.B32\
72 66 20 64 65 74 61 6C 73 6E 61 72 54 20 21 00048 P.AAF: .ASCII \! Translated from MACRO-32 by MARBLI V03\
20 79 62 20 32 33 2D 4F 52 43 41 40 20 60 6F 00057
33 30 56 20 49 4C 42 52 41 4D 00066
33 30 56 20 49 4C 31 30 30 2D 00070
00 00 4F 52 43 41 4D 2E 00074 P.AAG: .BLKB 0
00 00 53 5F 00074 P.AAH: .ASCII \.MACRO\<0><0>
00 00 53 5F 0007C P.AAI: .ASCII \_S\<0><0>
```

```
.PSECT S0WNS,NOEXE,2
```

```

00000 INPUT_RECORD:
00084 INPUT_PTR: .BLKB 132
00088 INPUT_LENGTH: .BLKB 4
0008C OUTPUT_RECORD:
00110 OUTPUT_PTR: .BLKB 132
00114 ARG_PTR: .BLKB 4
00118 ARG_LENGTH: .BLKB 4
0011C ARG_LIST: .BLKB 4096
0111C CALL_PTR: .BLKB 4
01120 CALL_LENGTH: .BLKB 4
01124 CALL_LIST: .BLKB 4096
02124 VAR_ARGS: .BLKB 4
02128 TERMINAL: .BLKB 4
```

```

0221C INPUT: .BLKB 244
02310 OUTPUT: .BLKB 244
0009 02404 $IOB$FILE_SPEC:
01 0E 02406 .WORD 9
00000000 02408 .BYTE 14, 1
001A 0240C $IOB$PROMPT:
01 0E 0240E .WORD 26
00000000 02410 .BYTE 14, 1
000B 02414 $IOB$DEFAULT:
01 0E 02416 .WORD 11
00000000 02418 .BYTE 14, 1
0015 0241C $IOB$PROMPT:
01 0E 0241E .WORD 21
00000000 02420 .BYTE 14, 1
0005 02424 $IOB$DEFAULT:
01 0E 02426 .WORD 5
00000000 02428 .BYTE 14, 1
002C 0242C $IOB$OUTPUT:
01 0E 0242E .WORD 44
00000000 02430 .BYTE 14, 1
0000 02434 $IOB$OUTPUT:
01 0E 02436 .WORD 0
00000C00 02438 .BYTE 14, 1
00000000 02438 .ADDRESS P.AAG

```

```

IOB$= TERMINAL
IOB$RESULTANT= TERMINAL+28
IOB$= INPUT
IOB$RESULTANT= INPUT+28
IOB$= OUTPUT
IOB$RESULTANT= OUTPUT+28
IOB$= TERMINAL
IOB$= TERMINAL
IOB$= INPUT
$IOB$FILE_SPEC= TERMINAL+52
IOB$= TERMINAL
IOB$= OUTPUT
$IOB$FILE_SPEC= TERMINAL+52
$IOB$RELATED= INPUT+28
IOB$= OUTPUT
IOB$= INPUT
IOB$= OUTPUT
IOB$= INPUT
IOB$= OUTPUT
IOB$= TERMINAL
.EXTRN XPOSOPEN, XPOSFAILURE
.EXTRN XSTSFREE TEMP, XPOSGET
.EXTRN XPOSIO FAILURE, XPOSPUT
.EXTRN XPOSCLOSE

```

.PSECT \$CODE\$,NOWRT,2

00F4	8F	00	SB 00000000G	0FFC 00000	CONVERT: .WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0132
			5A 00000000G	EF 9E 00002	MOVAB	XPOSGET, R11	
			59 00000000G	EF 9E 00009	MOVAB	XPOSOPEN, R10	
			58 00000000	EF 9E 00010	MOVAB	XPOSFAILURE, R9	
			58 0000	CF 9E 00017	MOVAB	IOBS, R8	
			5E	08 C2 0001C	SUBL	#8, SP	
			6E	00 2C 0001F	MOVCS	#0, (SP), #0, #244, IOBS	0155
			FE18	C8 0301003D	MOVL	#50397245, IOBS	
00F4	8F	00	FE36	C8 020E	MOVW	#526, IOBS\$RESULTANT+2	
			6E	00 2C 00039	MOVCS	#0, (SP), #0, #244, IOBS	0156
			FF0C	C8 0301003D	MOVL	#50397245, IOBS	
00F4	8F	00	FF2A	C8 020E	MOVW	#526, IOBS\$RESULTANT+2	
			6E	00 2C 00053	MOVCS	#0, (SP), #0, #244, IOBS	0157
			68	68 0005A	MOVL	#50397245, IOBS	
			1E	68 0301003D	MOVW	#526, IOBS\$RESULTANT+2	
			A8	020E	MOVAB	\$IOBS\$FILE_SPEC, IOBS\$+4	
			FE1C	C8 00F4	MOVAB	#1, IOBS\$+44	0159
			FE44	C8	PUSHL	R9	
				59 DD 00074	CLRL	-(SP)	
				7E D4 00076	PUSHAB	IOBS	
			6A	FE18	CALLS	#3, XPOSOPEN	
			50	FE3C	MOVL	IOBS\$+36, R0	0165
				03 FB 0007C	BEQL	2\$	
				09 13 00084	PUSHL	R0	
			00000000G	50 DD 00086	CALLS	#1, XSTSFREE TEMP	
			EF	01 FB 00088	MOVAB	\$IOBS\$PROMPT, -IOBS\$+36	
			FE3C	C8 00FC	MOVAB	#6, IOBS\$+44	
			FE44	C8	PUSHL	R9	
				59 DD 0009B	CLRL	-(SP)	
				7E D4 00090	PUSHAB	IOBS	
			6B	FE18	CALLS	#3, XPOSGET	
			4C	03 FB 000A3	BLBC	R0, 4\$	
			FF10	C8 FE4C	MOVAB	\$IOBS\$FILE_SPEC, IOBS\$+4	
			FF14	C8 0104	MOVAB	\$IOBS\$DEFAULT, IOBS\$+8	0172
			FF38	C8	MOVAB	#1, IOBS\$+44	
				01 90 000B7	PUSHAB	XPOSIO_FAILURE	
			00000000G	EF 9F 000BC	CLRL	-(SP)	
				7E D4 000C2	PUSHAB	IOBS	
				03 FB 000C8	CALLS	#3, XPOSOPEN	
			6A	FE18	BLBC	R0, 1\$	
			B1	03 FB 000CB	MOVL	IOBS\$+36, R0	0177
			50	FE3C	BEQL	3\$	
				50 E9 000D5	PUSHL	R0	
			00000000G	01 FB 000D7	CALLS	#1, XSTSFREE TEMP	
			FE3C	C8 010C	MOVAB	\$IOBS\$PROMPT, -IOBS\$+36	
			FE44	C8	MOVAB	#6, IOBS\$+44	
				06 90 000E5	PUSHL	R9	
				59 DD 000EA	CLRL	-(SP)	
				7E D4 000EC	PUSHAB	IOBS	
			6B	FE18	CALLS	#3, XPOSGET	
			03	03 FB C00F2	BLBS	IOBS	
				50 E8 000F5	BEQL	5\$	
				48:	PUSHL	R0	

04	A8	FE4C	015A	31 000F8	BRW	18\$	0185
08	A8	0114	C8 9E	000FB 5\$:	MOVAB	\$10BSFILE SPEC, IOBS\$+4	
0C	A8	FF28	C8 9E	00101	MOVAB	\$10BSDEFAULT, IOBS\$+8	
2E	A8		C8 9E	00107	MOVAB	\$10BSRELATED, IOBS\$+12	
2C	A8		02	88 0010D	BISB2	#2, IOBS\$+46	
			01	90 00111	MOVB	#1, IOBS\$+44	
			59	DD 00115	PUSHL	R9	
			7E	D4 00117	CLRL	-(SP)	
			58	DD 00119	PUSHL	R8	
44	6A	011C	C8 9E	0011B	CALLS	#3, XPOSOPEN	0188
2C	A8		07	90 00124	MOVAB	\$10BSOUTPUT, IOBS\$+68	
			59	DD 00128	MOVB	#7, IOBS\$+44	
			7E	D4 0012A	PUSHL	R9	
			58	DD 0012C	CLRL	-(SP)	
			03	FB 0012E	PUSHL	R8	
00000000G	EF	DD7C	C8 9E	00135	CALLS	#3, XPOSPUT	0191
DE00	C8		57	D4 0013C	MOVAB	OUTPUT_RECORD, OUTPUT_PTR	
FF38	C8		06	90 0013E	CLRL	PRINT COMMENTS	
			59	DD 00143	MOVB	#6, IOBS\$+44	
			7E	D4 00145	PUSHL	R9	
		FF0C	C8 9F	00147	CLRL	-(SP)	
	68		03	FB 0014B	PUSHAB	IOBS\$	
	03		50	E8 0014E	CALLS	#3, XPOSGET	
			31	00151	BLBS	R0, 8\$	
DD78	C8	FF40	C8 3C	00154	BRW	17\$	0201
DD74	C8	FF44	C8 D0	00158	MOVZWL	INPUT+52, INPUT_LENGTH	
	50	DD78	C8 D0	00162	MOVL	INPUT+56, INPUT_PTR	
			E8	19 00167	MOVL	INPUT_LENGTH, R0	
			1C	12 00169	BLSS	7\$	
			57	E9 0016B	BNEQ	10\$	
44	D0	0124	C8 9E	0016E	BLBC	PRINT COMMENTS, 6\$	0206
2C	A8		07	90 00174	MOVAB	\$10BSOUTPUT, IOBS\$+68	
			59	DD 00178	MOVB	#7, IOBS\$+44	
			7E	D4 0017A	PUSHL	R9	
			58	DD 0017C	CLRL	-(SP)	
			03	FB 0017E	PUSHL	R8	
00000000G	EF		B7	11 00185	CALLS	#3, XPOSPUT	0208
			DE00	DO 00187	BRB	6\$	
0082	8F	00 DD74	56 D8	10\$:	MOVL	OUTPUT_PTR, R6	0213
				50 2C 0018C	MOVS	R0, @INPUT_PTR, #0, #130, (R6)	
DD7C	C8	0083	8F	66 00195	LOCC	#0, #131, OUTPUT_RECORD	0215
				00 3A 00196	BNEQ	11\$	
				02 12 0019E	CLRL	R1	
DD78	C8		50 51	D4 001A0	MOVAB	OUTPUT_RECORD, R0	0217
			51 3B	C8 9E 001A2	11\$:	SUBL3	R0, R1- INPUT_LENGTH
				50 C3 001A7	CMPB	(R6), #59	
			66 91	001AD	BNEQ	13\$	
			3C 12	001B0	MOVB	#33, (R6)	
			21 90	001B2	BLBC	PRINT COMMENTS, 12\$	
			57 E9	001B5	MOVW	INPUT_LENGTH, \$10BSOUTPUT	
02	AE	DD78	C8 80	001B8	MOVW	#14, \$10BSOUTPUT+2	0224
03	AE		0E	90 001BD	MOVW	#1, \$10BSOUTPUT+3	
04	AE		01	90 001C1	MOVW	R6, \$10BSOUTPUT+4	
44	A8		56	00 001C5	MOVL	\$10BSOUTPUT, IOBS\$+68	
2C	A8		6E	9E 001C9	MOVAB	#7, IOBS\$+44	
			07	90 001CD	MOVB	R9	
			59	DD 001D1	PUSHL		

				7E	D4	001D3	CLRL	-(SP)				
				58	DD	001D5	PUSHL	R8				
				03	FB	001D7	CALLS	#3, XPOS\$PUT				
				C8	DD	001DE	MOVL	OUTPUT_PTR, R0				
				A0	91	001E3	CMPB	1(R0), #42				
				9C	12	001E7	BNEQ	9S				
				01	DD	001E9	MOVL	#1, PRINT_COMMENTS				
				97	11	001EC	BRB	9S				
66	DD78	C8	0000'	CF	06	39	001EE	12S:	MATCHC	#6, P.AAH, INPUT_LENGTH, (R6)		
				53	03	13	001F7		BEQL	14S		
				53	06	DD	001F9		MOVL	#6, R3		
				53	06	C2	001FC	14S:	SUBL2	#6, R3		
66	DD78	C8	0000'	CF	84	13	001FF		BEQL	9S		
				53	02	39	00201		MATCHC	#2, P.AAI, INPUT_LENGTH, (R6)		
				53	03	13	0020A		BEQL	15S		
				53	02	DD	0020C		MOVL	#2, R3		
				53	02	C2	0020F	15S:	SUBL2	#2, R3		
			0000V	CF	05	13	00212		BEQL	16S		
			FF38	C8	00	FB	00214		CALLS	#0, CONVMACRO		
				FF22	31	00219	16S:		BRW	6S		
				FF0C	02	90	0021C	17S:	MOVB	#2, IOBS+44		
				2C	59	DD	00221		PUSHL	R9		
				A8	7E	D4	00223		CLRL	-(SP)		
					C8	9F	00225		PUSHAB	IOBS		
					03	FB	00229		CALLS	#3, XPOS\$CLOSE		
					02	90	00230		MOVB	#2, IOBS+44		
					59	DD	00234		PUSHL	R9		
					7E	D4	00236		CLRL	-(SP)		
					58	DD	00238		PUSHL	R8		
					03	FB	0023A		CALLS	#3, XPOS\$CLOSE		
					02	90	00241		MOVB	#2, IOBS+44		
					59	DD	00246		PUSHL	R9		
					7E	D4	00248		CLRL	-(SP)		
					C8	9F	0024A		PUSHAB	IOBS		
					03	FB	0024E		CALLS	#3, XPOS\$CLOSE		
					50	00208001	8F	00255	18S:	MOVL	#2129921, R0	
							04	0025C		RET		

: Routine Size: 605 bytes, Routine Base: \$CODE\$ + 0000

```
256 0254 1 ROUTINE CONVMACRO: NOVALUE=
257 0255 1
258 0256 1 !++
259 0257 1 ! FUNCTIONAL DESCRIPTION:
260 0258 1 ! This routine reads and processes a MARS system macro
261 0259 1 ! It calls the routine CONVARGLIST to convert the argument list
262 0260 1 ! of the macro definition.
263 0261 1
264 0262 1 ! PARAMETERS:
265 0263 1
266 0264 1 ! NONE
267 0265 1
268 0266 1 ! ROUTINE VALUE:
269 0267 1
270 0268 1 ! NONE
271 0269 1
272 0270 1 !--
273 0271 1
274 0272 2 BEGIN
275 0273 2 LOCAL
276 0274 2 PREFIX_PTR,
277 0275 2 PTR, ! Temp CH$ pointer for copying arglists
278 0276 2 TEMP_PTR1,
279 0277 2 TEMP_PTR2,
280 0278 2 TEMPSTRING: VECTOR[CH$ALLOCATION(80)],
281 0279 2 MACRO_PTR,
282 0280 2 MACRO_LENGTH,
283 0281 2 MACRO_NAME: VECTOR[CH$ALLOCATION(31)];
284 0282 2
285 0283 2
286 0284 2 MACRO_PTR = CH$PTR(MACRO_NAME);
287 0285 2
288 0286 2
289 0287 2 ! If the .MACRO line contains the string 'CJFS', then change the routine
290 0288 2 ! name prefix from SYSS to CJFS. Do the same for RUF.
291 0289 2
292 0290 2 PREFIX_PTR = CH$PTR(UPLIT BYTE('SYS'));
293 0291 2 IF CH$FIND SUB(MAX REC SIZE, .OUTPUT_PTR, CH$LEN_PTR('CJFS')) NEQ 0
294 0292 2 THEN PREFIX_PTR = CH$PTR(UPLIT BYTE('CJF'));
295 0293 2 IF CH$FIND SUB(MAX REC SIZE, .OUTPUT_PTR, CH$LEN_PTR('RUF')) NEQ 0
296 0294 2 THEN PREFIX_PTR = CH$PTR(UPLIT BYTE('RUF'));
297 0295 2
298 0296 2
299 0297 2 ! Search for the '$' which begins the macro name and
300 0298 2 ! the '_' which ends it, and initialize MACRO_NAME and MACRO_LENGTH
301 0299 2
302 0300 2 TEMP_PTR1 = CH$FIND CH(MAX_REC_SIZE, .OUTPUT_PTR, XC'$');
303 0301 2 TEMP_PTR2 = CH$FIND CH(MAX_REC_SIZE, .OUTPUT_PTR, XC'_');
304 0302 2 MACRO_LENGTH = CH$DIFF(.TEMP_PTR2, .TEMP_PTR1);
305 0303 2 CH$MOVE(.MACRO_LENGTH, .TEMP_PTR1, .MACRO_PTR);
306 0304 2
307 0305 2
308 0306 2 ! Inform user of progress
309 0307 2 !
310 0308 2 ! CH$COPY(CH$LEN_PTR('Macro: '),
311 0309 2 ! MACRO_LENGTH, .MACRO_PTR,
312 0310 2 !
```

```
313 0311 2 :      80, CH$PTR(TEMPSTRING) );  
314 0312 2 :  
315 0313 2 :      $XPO_PUT( IOB=TERMINAL, STRING=(80, CH$PTR(TEMPSTRING) ) );  
316 0314 2 :  
317 0315 2 :  
318 0316 2 :      ! Gather the argument list  
319 0317 2 :  
320 0318 2 :      CONVARGLIST(.TEMP_PTR2);  
321 0319 2 :  
322 0320 2 :  
323 0321 2 :      ! Put out one of the lines  
324 0322 2 :      'KEYWORDMACRO $name(formal=default,...)='  
325 0323 2 :      or  
326 0324 2 :      'MACRO $name[]='      if VAR_ARGS is true  
327 0325 2 :  
328 0326 2 :  
329 0327 2 :      IF .VAR_ARGS  
330 0328 2 :      THEN  
331 0329 2 :          CHSCOPY(  
332 0330 2 :              CH$LEN_PTR('MACRO '),  
333 0331 2 :              .MACRO_LENGTH,  
334 0332 2 :              .MACRO_PTR,  
335 0333 2 :              CH$LEN_PTR('[]='),  
336 0334 2 :              0,  
337 0335 2 :              MAY_REC_SIZE,  
338 0336 2 :              .OUTPUT_PTR)  
339 0337 3 :  
340 0338 3 :      ELSE  
341 0339 3 :          BEGIN  
342 0340 3 :          LOCAL  
343 0341 3 :              PTR;  
344 0342 3 :  
345 0343 3 :      PTR = CHSMOVE(CH$LEN_PTR('KEYWORDMACRO '), .OUTPUT_PTR);  
346 0344 3 :      PTR = CHSMOVE(.MACRO_LENGTH, .MACRO_PTR, .PTR);  
347 0345 3 :  
348 0346 4 :      IF .ARG_LENGTH GTR 0      ! If there are formal parameters, then  
349 0347 4 :      THEN      ! put out the argument list.  
350 0348 4 :          BEGIN  
351 0349 4 :              PTR = CHSMOVE( CH$LEN_PTR('('), .PTR );  
352 0350 4 :              PTR = OUTLIST( .PTR, .ARG_LENGTH, .ARG_PTR );  
353 0351 4 :              PTR = CHSMOVE( CH$LEN_PTR(')='), .PTR )  
354 0352 3 :  
355 0353 3 :      ELSE  
356 0354 3 :          CH$WCHAR_A(%C'=', PTR);      ! There were no formal parameters  
357 0355 2 :  
358 0356 2 :  
359 0357 2 :  
360 0358 2 :      $XPO_PUT( IOB=OUTPUT,  
361 0359 2 :                      STRING=(TRUNCATED_OUTPUT, .OUTPUT_PTR) );  
362 0360 2 :  
363 0361 2 :  
364 0362 2 :      ! Put out the line "  ("  
365 0363 2 :  
366 0364 2 :      $XPO_PUT( IOB=OUTPUT, STRING=( '  (' ) );  
367 0365 2 :  
368 0366 2 :  
369 0367 2 :      ! Put out the line "  EXTERNAL ROUTINE SYSS$name: BLISS ADDRESSING_MODE(GENERAL);"
```

```

370 0368 2
371 0369 2  CH$COPY(
372 0370 2    CH$LEN_PTR('  EXTERNAL ROUTINE '),
373 0371 2    3, .PREFIX_PTR,
374 0372 2    .MACRO_LENGTH,
375 0373 2    .MACRO_PTR,
376 0374 2    CH$LEN_PTR(': BLISS ADDRESSING_MODE(GENERAL);'),
377 0375 2    0,
378 0376 2    MAX_REC_SIZE,
379 0377 2    .OUTPUT_PTR);
380 0378 2    $XPO_PUT( IOB=OUTPUT, STRING=(TRUNCATED_OUTPUT, .OUTPUT_PTR) );
381 0379 2
382 0380 2
383 0381 2  ! Put out the line "  SYSSname(formal,...)"
384 0382 2
385 0383 2  PTR = CH$COPY(
386 0384 2    CH$LEN_PTR('  '),
387 0385 2    3, .PREFIX_PTR,
388 0386 2    .MACRO_LENGTH,
389 0387 2    .MACRO_PTR,
390 0388 2    CH$LEN_PTR('('),
391 0389 2    0,
392 0390 2    .MACRO_LENGTH + 8, ! Exact copying
393 0391 2    .OUTPUT_PTR);
394 0392 2
395 0393 2  PTR = OUTLIST( .PTR, .CALL_LENGTH, .CALL_PTR);
396 0394 2  PTR = CH$MOVE( CH$LEN_PTR(')'), XCHAR(0)-), .PTR );
397 0395 2
398 0396 2
399 0397 2  $XPO_PUT( IOB=OUTPUT, STRING=(TRUNCATED_OUTPUT, .OUTPUT_PTR) );
400 0398 2
401 0399 2
402 0400 2  ! Close up the macro declaration
403 0401 2
404 0402 3  $XPO_PUT( IOB=OUTPUT, STRING=(    ) %:' ) )
405 0403 1  END;

```

										.PSECT	SPLIT\$,NOWRT,NOEXE,2									
00	00	20	4F	52	43	41	4D	44	52	4F	53	59	53	00080 P.AAJ:	.ASCII \SYS\					
											00083			.BLKB 1						
											24	46	4A	43	00084 P.AAK:	.ASCII \CJFS\				
											46	4A	43	00088 P.AAL:	.ASCII \CJF\					
											0008B			.BLKB 1						
											24	46	55	52	0008C P.AAM:	.ASCII \RUF\\$\				
											46	55	52	00090 P.AAN:	.ASCII \RUF\					
											00093			.BLKB 1						
											00	00	20	4F	52	43	41	4D	00094 P.AAO:	.ASCII \MACRO \<0><0>
											00	3D	5D	5B	0009C P.AAP:	.ASCII \[]=\<0>				
											000A0 P.AAQ:	.ASCII \KEYWORDMACRO \<0><0><0>								
											00	00	00	28	000B0 P.AAR:	.ASCII \(\<0><0><0>				
											00	00	3D	29	000B4 P.AAS:	.ASCII \)=\<0><0>				
											28	20	20	20	000B8 P.AAT:	.ASCII \ (\				
											000BD			.BLKB 3						

```

4F 52 20 4C 41 4E 52 45 54 58 45 20 20 20 000C0 P.AAU: .ASCII \ EXTERNAL ROUTINE \<0><0><0>
53 53 45 52 44 44 41 20 53 53 49 4C 42 20 3A 000D8 P.AAV: .ASCII \: BLISS ADDRESSING_MODE(GENERAL);\<0><0>
41 52 45 4E 45 47 28 45 44 4F 4D 5F 47 4E 49 000E7
00 00 3B 29 4C 000F6
00 000FB .ASCII <0>
20 20 20 000FC P.AAW: .ASCII \
00 00 00 28 00100 P.AAX: .ASCII \(\<0><0><0>
00 00 00 29 00104 P.AAY: .ASCII \)\<0><0><0>
00 00 00 00 00108 P.AAZ: .ASCII <0><0><0><0>
3B 25 20 29 20 20 20 0010C P.AAZ: .ASCII \ ) %;\

.PSECT $0WN$,NOEXE,2

0005 0243C $IOB$OUTPUT:
01 0E 0243E .WORD 5
00000000 02440 .BYTE 14, 1
0008 02444 $IOB$OUTPUT:
00000000 02440 .ADDRESS P.AAT
0008 02444 $IOB$OUTPUT:
01 0E 02446 .WORD 8
00000000 02443 .BYTE 14, 1
00000000 02443 .ADDRESS P.AAZ

IOB$= OUTPUT
IOB$= OUTPUT
IOB$= OUTPUT
IOB$= OUTPUT
IOB$= OUTPUT

```

56	57	51	D4	0005D	51	D0	0005F	68:	CLRL	R1		
69	57	52	C3	00062	52	C3	00062		MOVL	R1, TEMP_PTR2	0302	
	62	56	28	00066	56	28	00066		SUBL3	TEMP_PTR1, TEMP_PTR2, MACRO_LENGTH	0303	
	0000V	CF	01	F8	0006C	57	DD	0006A		MOVCL3	MACRO_LENGTH, (TEMP_PTR1), (MACRO_PTR)	0318
	2F	2014	CB	E9	00071				PUSHL	TEMP_PTR2		
	58	82	8F	9A	00076				CALLS	#1, CONVARGLIST	0326	
	57	6B	D0	0007A				BLBC	VAR_ARGS, 78	0328		
58	00	0000'	CF	06	2C	0007D			MOVZBL	#130, R8	0335	
				67		00084			MOVLS	OUTPUT_PTR, R7		
					4F	18	00085		BGEQ	#6, R7		
58	00	69	57	06	CO	00087			ADDL2	#6, R8		
			58	06	C2	0008A			SUBL2	MACRO_LENGTH, (MACRO_PTR), #0, R8, (R7)		
58	00	0000'	CF	56	2C	0008D			MOVCS			
				67		00092						
					41	18	00093		BGEQ	10\$		
			57	56	CO	00095			ADDL2	MACRO_LENGTH, R7		
58	00	0000'	CF	58	C2	00098			SUBL2	MACRO_LENGTH, R8		
				03	2C	0009B			MOVCS	#3, P.AAP, #0, R8, (R7)		
				67		000A2						
00	BB	0000'	CF	31	11	000A3			BRB	10\$	0328	
63	69			0D	28	000A5	78:		MOVCL3	#13, P.AAQ, @OUTPUT_PTR	0341	
	50	08		56	28	000AC			MOVCL3	MACRO_LENGTH, (MACRO_PTR), (PTR)	0342	
	83	0000'	CF	A8	D0	000B0			MOVL	ARG_LENGTH, R0	0344	
		04		18	15	000B4			BLEQ	8\$		
				AB	90	00086			MOVB	P.AAR, (PTR)+	0347	
				AB	DD	0008B			PUSHL	ARG_PTR	0348	
				50	DD	000BE			PUSHL	R0		
				53	DD	000C0			PUSHL	PTR		
	0000V	CF		03	FB	000C2			CALLS	#3, OUTLIST		
			53	50	D0	000C7			MOVL	R0, PTR		
			83	0000'	CF	B0	000CA		MOVW	P.AAS, (PTR)+	0349	
				03	11	000CF			BRB	9\$	0352	
			83	3D	90	000D1	88:		MOVB	#61, (PTR)+		
				63	94	000D4	98:		CLRB	(PTR)		
FF7C	CB	0083	8F	00	3A	000D6	108:		LOCC	#0, #131, OUTPUT_RECORD	0354	
				02	12	000DE			BNEQ	11\$	0359	
				51	D4	000E0			CLRL	R1		
6E	50	FF7C		CB	9E	000E2	118:		MOVAB	OUTPUT_RECORD, R0		
	51			50	A3	000E7			SUBL3	R0, R1, \$IOBS\$OUTPUT		
	02	AE		0E	90	000EB			MOVB	#14, \$IOBS\$OUTPUT+2		
	03	AE		01	90	000EF			MOVB	#1, \$IOBS\$OUTPUT+3		
	04	AE		68	D0	000F3			MOVL	OUTPUT_PTR, \$IOBS\$OUTPUT+4		
	2244	CB		6E	9E	000F7			MOVAB	\$IOBS\$OUTPUT, IOBS+68		
	222C	CB		07	90	000FC			MOVB	#7, IOBS+44		
				EF	9F	00101			PUSHAB	XP0\$FAILURE		
				7E	D4	00107			CLRL	-(SP)		
				2200	CB	9F	00109		PUSHAB	IOBS		
00000000G	EF			03	FB	0010D			CALLS	#3, XPOSPUT		
2244	CB			CB	9E	00114			MOVAB	\$IOBS\$OUTPUT, IOBS+68	0364	
222C	CB			07	90	0011B			MOVB	#7, IOBS+44		
				00000000G	EF	9F	00120		PUSHAB	XP0\$FAILURE		
				7E	D4	00126			CLRL	-(SP)		
				2200	CB	9F	00128		PUSHAB	IOBS		
00000000G	EF			03	FB	0012C			CALLS	#3, XPOSPUT		
	58	82		8F	9A	00133			MOVZBL	#130, R8	0369	

58	00	0000'	57	68	00	00137	MOVL	OUTPUT_PTR, R7	0377	
				15	2C	0013A	MOVCS	#21, P-AAU, #0, R8, (R7)		
				67		00141	BGEQ	12\$		
			57	2A	18	00142	ADDL2	#21, R7		
58	00	6A	58	15	C0	00144	SUBL2	#21, R8		
			03	C2	00147	MOVCS	#3, (PREFIX_PTR), #0, R8, (R7)			
			67	03	2C	0014A	BGEQ	12\$		
			67	1C	18	00150	ADDL2	#3, R7		
58	00	69	57	03	C0	00152	SUBL2	#3, R8		
			58	C2	00155	MOVCS	MACRO_LENGTH, (MACRO_PTR), #0, R8, (R7)			
			56	56	2C	00158	BGEQ	12\$		
			67	67		0015D	ADDL2	MACRO_LENGTH, R7		
58	00	0000'	57	0E	18	0015E	SUBL2	MACRO_LENGTH, R8		
			58	56	C0	00160	MOVCS	#33, P-AAV, #0, R8, (R7)		
			21	56	C2	00163	BGEQ	12\$		
58	00	0000'	57	21	2C	00166	ADDL2	MACRO_LENGTH, R7		
			58	67		0016D	SUBL2	MACRO_LENGTH, R8		
	FF7C	CB	0083	8F	00	3A	0016E	MOVCS	#33, P-AAV, #0, R8, (R7)	
				02	12	00176	12\$:	LOC	0, #131, OUTPUT_RECORD	
				51	D4	00178	BNEQ	13\$		
	6E	50	FF7C	CB	9E	0017A	CLRL	R1		
		51		50	A3	0017F	MOVAB	OUTPUT_RECORD, R0		
		02	AE	0E	90	00183	SUBW3	R0, R1, \$IOBS\$OUTPUT		
		03	AE	01	90	00187	MOVB	#1, \$IOBS\$OUTPUT+2		
		04	AE	6B	00	0018B	MOVB	#1, \$IOBS\$OUTPUT+3		
	2244	CB	222C	CB	6E	9E	0018F	MOVL	OUTPUT_PTR, \$IOBS\$OUTPUT+4	
		07		07	90	00194	MOVAB	\$IOBS\$OUTPUT, IOBS+68		
		00000000G		EF	9F	00199	MOVB	#7, IOBS+44		
				7E	D4	0019F	PUSHAB	XPOSFAILURE		
				CB	9F	001A1	CLRL	-(SP)		
	00000000G	EF	2200	03	FB	001A5	PUSHAB	IOBS		
		58		A6	9E	001AC	CALLS	#3, XPOSPUT	0390	
58	00	0000'	57	6B	00	001B0	MOVAB	8(R6), R8	0391	
		57		04	2C	001B3	MOVL	OUTPUT_PTR, R7		
		04		67		001BA	MOVCS	#4, P-AAW, #0, R8, (R7)		
				2A	18	001BB	BGEQ	14\$		
			57	04	C0	001BD	ADDL2	#4, R7		
58	00	6A	58	04	C2	001C0	SUBL2	#4, R8		
			03	03	2C	001C3	MOVCS	#3, (PREFIX_PTR), #0, R8, (R7)		
			67	67		001C8	BGEQ	14\$		
			57	1C	18	001C9	ADDL2	#3, R7		
			58	03	C0	001CB	SUBL2	#3, R8		
58	00	69	58	03	C2	001CE	MOVCS	MACRO_LENGTH, (MACRO_PTR), #0, R8, (R7)		
			57	56	2C	001D1	BGEQ	14\$		
			58	67		001D6	ADDL2	MACRO_LENGTH, R7		
58	00	0000'	57	0E	18	001D7	SUBL2	MACRO_LENGTH, R8		
			58	56	C0	001D9	MOVCS	#1, P-AAX, #0, R8, (R7)		
	0000V	CF	58	56	C2	001DC	BGEQ	14\$		
58	00	0000'	57	01	2C	001DF	ADDL2	MACRO_LENGTH, R7		
		58	67	67		001E6	SUBL2	MACRO_LENGTH, R8		
		0000V	CF	1000	CB	DD	001E7	MOVCS	#1, P-AAX, #0, R8, (R7)	
			1010	CB	DD	001EB	14\$:	PUSHL	CALL_PTR	
				53	DD	001EF	PUSHL		CALL_LENGTH	
				03	FB	001F1	PUSHL		PTR	
				50	DO	001F6	CALLS	#3, OUTLIST		
				83	0000'	CF	MOVW	R0, PTR		
					BO	001F9	MOVW	F-AAV, (PTR)+		

FF7C	CB	0083	8F	00	3A	001FE	LOCC	#0	#131, OUTPUT_RECORD	0397
				02	12	00206	BNEQ	15\$		
				51	D4	00208	CLRL	R1		
6E		50	FF7C	CB	9E	0020A	15\$:	MOVAB	OUTPUT_RECORD, R0	
		51		50	A3	0020F	SUBW3	RO	R1-\$IOBS\$OUTPUT	
	02	AE		0E	90	00213	MOVAB	#14	\$IOBS\$OUTPUT+2	
	03	AE		01	90	00217	MOVAB	#1	\$IOBS\$OUTPUT+3	
	04	AE		6B	D0	0021B	MOVL	OUTPUT_PTR	\$IOBS\$OUTPUT+4	
2244	CB			6E	9E	0021F	MOVAB	\$IOBS\$OUTPUT	, IOBS\$+68	
222C	CB			07	90	00224	MOVAB	#7	, IOBS\$+44	
				EF	9F	00229	PUSHAB	XPOSFAILURE		
				7E	D4	0022F	CLRL	-(SP)		
				CB	9F	00231	PUSHAB	IOBS		
00000000G	EF		2200	03	FB	00235	CALLS	#3	, XPOSPUT	
2244	CB		2334	CB	9E	0023C	MOVAB	\$IOBS\$OUTPUT	, IOBS\$+68	
222C	CB			07	90	00243	MOVAB	#7	, IOBS\$+44	
				EF	9F	00248	PUSHAB	XPOSFAILURE		
				7E	D4	0024E	CLRL	-(SP)		
				CB	9F	00250	PUSHAB	IOBS		
00000000G	EF		2200	03	FB	00254	CALLS	#3	, XPOSPUT	
				04	0025B		RET			0403

; Routine Size: 604 bytes, Routine Base: \$CODE\$ + 0250

```

407 0404 1 ROUTINE OUTLIST(BUFPTR, LEN, LPTR)=
408 0405 1
409 0406 1 FUNCTION
410 0407 1 Dump an argument list buffer (which may span several source
411 0408 1 records..) to the output file. Continuation lines are flagged
412 0409 1 by an ASCII NUL (0) byte.
413 0410 1 INPUTS
414 0411 1 BUFPTR - byte-pointer into the OUTPUT_RECORD containing prefix
415 0412 1 information for the first record written.
416 0413 1
417 0414 1 LEN - Length of the argument-list string.
418 0415 1
419 0416 1 OUTPUT LPTR - Pointer to the argument-list string.
420 0417 1
421 0418 1 Updated 'BUFPTR'
422 0419 1
423 0420 2 BEGIN
424 0421 2 LOCAL
425 0422 2 IPTR,          ! Input string pointer (arg-list)
426 0423 2 OPTR,          ! Copy of BUFPTR...
427 0424 2 CHAR;          ! Character temporary
428 0425 2
429 0426 2 IPTR = .LPTR;
430 0427 2 OPTR = .BUFPTR;
431 0428 2
432 0429 2 DECR I FROM .LEN-1 TO 0 DO
433 0430 3 BEGIN
434 0431 3 CHAR = CHSRCHAR_A( IPTR );      ! Get next character and copy to
435 0432 3 CHSWCHAR_A( .CHAR, OPTR );      ! output string
436 0433 3
437 0434 3 IF .CHAR EQL 0
438 0435 3 THEN
439 0436 4 BEGIN
440 0437 4 $XPO_PUT( IOB=OUTPUT,
441 0438 4           STRING=( TRUNCATED_OUTPUT, .OUTPUT_PTR)
442 0439 4           );
443 0440 4           OPTR = .OUTPUT_PTR;          ! Restore buffer ptr.
444 0441 4           CHSWCHAR_A( XCHAR(9), OPTR )      ! and TAB from left-margin
445 0442 4           END
446 0443 2 END;
447 0444 2
448 0445 2 .OPTR
449 0446 1 END;

```

JOBS= OUTPUT

5E	003C	00000	OUTLIST: .WORD	Save R2, R3, R4, R5	: 0404
55	08	C2 00002	SUBL2	#8, SP	
52	0C	AC D0 00005	MOVL	LPTR, IPTR	: 0426
	04	AC 7D 00009	MOVQ	BUFPTR, OPTR	: 0427
		52 11 00000	BRB	3S	: 0429
54	85	9A 0000F	1\$: MOVZBL	(IPTR)+, (CHAR	: 0431
82	54	90 00012	MOVBL	CHAR, (OPTR)+	: 0432
	54	D5 00015	TSTL	CHAR	: 0434

0000' CF	0083 8F	48 12 00017	BNEQ	3\$	0439
		00 3A 00019	LOCC	#0, #131, OUTPUT_RECORD	
		02 12 00021	BNEQ	2\$	
		51 D4 00023	CLRL	R1	
6E	50 0000'	CF 9E 00025	2\$:	MOVAB	OUTPUT_RECORD, R0
	51	50 A3 0002A	SUBW3	R0 R1-\$IOBS\$OUTPUT	
	02 AE	0E 90 0002E	MOV8	#14, \$IOBS\$OUTPUT+2	
	03 AE	01 90 00032	MOV8	#1, \$IOBS\$OUTPUT+3	
	04 AE	0000' CF D0 00036	MOVL	OUTPUT_PTR, \$IOBS\$OUTPUT+4	
	0000' CF	6E 9E 0003C	MOVAB	\$IOBS\$OUTPUT, IOBS+68	
	0000' CF	07 90 00041	MOV8	#7, IOBS+44	
	00000000G	EF 9F 00046	PUSHAB	XPOSFAILURE	
		7E D4 0004C	CLRL	-(SP)	
	0000' CF 9F 0004E	PUSHAB	IOBS		
00000000G	EF 0000' CF FB 00052	CALLS	#3, XPOSPUT		
	52 0000' CF D0 00059	MOVL	OUTPUT_PTR, OPTR		0440
	82 09 90 0005E	MOV8	#9, (OPTR)+		C441
	AB 53 F4 00061	S0BGEQ	I, 1\$		0434
	50 52 D0 00064	MOVL	OPTR, R0		0446
	04 00067	RET			

; Routine Size: 104 bytes, Routine Base: \$CODE\$ + 04B9

```
451 0447 1 ROUTINE CONVARGLIST(TEMP_PTR): NOVALUE=
452 0448 1
453 0449 1 !++
454 0450 1 !++
455 0451 1 !++
456 0452 1 !++
457 0453 1 !++
458 0454 1 !++
459 0455 1 !++
460 0456 1 !++
461 0457 1 !++
462 0458 1 !++
463 0459 1 !++
464 0460 1 !++
465 0461 1 !++
466 0462 1 !++
467 0463 1 !++
468 0464 1 !++
469 0465 1 !++
470 0466 1 !++
471 0467 1 !++
472 0468 1 !++
473 0469 1 !++
474 0470 1 !++
475 0471 1 !++
476 0472 1 !++
477 0473 1 !++
478 0474 1 !++
479 0475 2 !++
480 0476 2 !++
481 0477 2 !++
482 0478 2 !++
483 0479 2 !++
484 0480 2 !++
485 0481 2 !++
486 0482 2 !++
487 0483 2 !++
488 0484 2 !++
489 0485 2 !++
490 M 0486 2 !++
491 0487 2 !++
492 0488 2 !++
493 0489 2 !++
494 0490 2 !++
495 0491 2 !++
496 0492 2 !++
497 0493 2 !++
498 0494 2 !++
499 0495 2 !++
500 0496 2 !++
501 0497 2 !++
502 0498 2 !++
503 0499 2 !++
504 0500 2 !++
505 0501 2 !++
506 0502 2 !++
507 0503 2 !++
```

FUNCTIONAL DESCRIPTION:
This routine gathers the argument list of a macro definition.
It processes continuation lines. It eliminates unnecessary
characters and places the arguments with default values in ARG_LIST
and a list of the formal names in CALL_LIST

PARAMETERS:

TEMP_PTR = The value of a CH pointer in the macro
definition line.

IMPLICIT OUTPUTS:

ARG_LIST	= String of formal arguments with default values
ARG_PTR	= String pointer to ARG_LIST
ARG_LENGTH	= Length of string in ARG_LIST
CALL_LIST	= String of formal arguments
CALL_PTR	= String pointer to CALL_LIST
CALL_LENGTH	= Length of string in CALL_LIST
VAR_ARGS	= True if the last formal argument was END_VARNUM_ARGS

ROUTINE VALUE:
NONE

--

BEGIN
LOCAL
LIST_PTR, ! Input character string pointer
CHAR, ! Character from string
STATE. ! State of parse
0 = No argument seen yet
1 = Scanning formal name
2 = Scanning default value

BEGIN_ARG: ! Character string pointer to beginning of argument

MACRO
TRY_ADD_DEFAULT=
IF :STATE EQL 1 THEN (CH\$WCHAR_A(%C'=', ARG_PTR); CH\$WCHAR_A(%C'0', ARG_PTR)) %;

! Initialize
ARG_PTR = CH\$PTR(ARG_LIST);
CH\$FILL(0, MAX_ARG_LIST, .ARG_PTR);
CALL_PTR = CH\$PTR(CALL_LIST);
CH\$FILL(0, MAX_ARG_LIST, .CALL_PTR);
LIST_PTR = CH\$PLUST(TEMP_PTR, 3);
VAR_ARGS = 0;
STATE = 0;

! Scan the argument list

```

: 508 0504 2 ; Exit when an end of line (not preceded by line continuation mark) is read
: 509 0505 2
: 510 0506 2 REPEAT
: 511 0507 2 BEGIN
: 512 0508 2 CHAR = CH$RCHAR_A(list_ptr);
: 513 0509 2 SELECTONE .CHAR_OF
: 514 0510 2 SET
: 515 0511 2
: 516 0512 3 [0]: BEGIN
: 517 0513 4
: 518 0514 4
: 519 0515 4 ; End of list
: 520 0516 4 ; Add a default value to the last argument if it did not
: 521 0517 4 ; have one.
: 522 0518 4 ; Set VAR_ARGS if the name of the last argument is
: 523 0519 4 ; END_VARNUM_ARGS
: 524 0520 4
: 525 0521 4 TRY_ADD_DEFAULT:
: 526 0522 4 IF .STATE GEQ 1
: 527 0523 4 THEN
: 528 0524 4 IF
: 529 0525 4 CH$EQL(
: 530 0526 4 CH$DIFF(.CALL_PTR, .BEGIN_ARG),
: 531 0527 4 .BEGIN_ARG,
: 532 0528 4 CH$LEN_PTR('END_VARNUM_ARGS'),
: 533 0529 4 0)
: 534 0530 4 THEN
: 535 0531 4 VAR_ARGS = 1;
: 536 0532 4
: 537 0533 4 EXITLOOP
: 538 0534 3 END;
: 539 0535 3
: 540 0536 3 [%C'A' TO %C'Z', %C'0' TO %C'9', %C'_', %C'$']:
: 541 0537 4 BEGIN
: 542 0538 4
: 543 0539 4 ; Symbol constituent
: 544 0540 4 ; If STATE is zero, it is the first character of a formal name
: 545 0541 4 ; STATE distinguishes scanning formal name from scanning default value
: 546 0542 4
: 547 0543 4 IF .STATE EQ 0
: 548 0544 4 THEN
: 549 0545 5 BEGIN
: 550 0546 5 STATE = 1;
: 551 0547 5 BEGIN_ARG = .CALL_PTR
: 552 0548 4 END;
: 553 0549 4
: 554 0550 4 IF .STATE EQ 1 THEN CH$WCHAR_A(.CHAR, CALL_PTR);
: 555 0551 4 CH$WCHAR_A(.CHAR, ARG_PTR)
: 556 0552 3 END;
: 557 0553 3
: 558 0554 3 [%C'=']:
: 559 0555 4 BEGIN
: 560 0556 4
: 561 0557 4 ; Beginning of default value
: 562 0558 4 ; Set STATE to indicate scanning default and collect character
: 563 0559 4
: 564 0560 4 STATE = 2;

```

```
565      0561 4      CHSWCHAR_A(.CHAR, ARG_PTR)
566      0562 3      END;
567      0563 3
568      0564 3      [%C',']:
569      0565 4      BEGIN
570      0566 4      | End of argument
571      0567 4      | Add a default value to the argument if it did not have one
572      0568 4      | Collect the character
573      0569 4      | Reset STATE to indicate not within an argument
574      0570 4
575      0571 4
576      0572 4      TRY ADD DEFAULT:
577      0573 4      CHSWCHAR_A(.CHAR, CALL_PTR);
578      0574 4      CHSWCHAR_A(.CHAR, ARG_PTR);
579      0575 4      STATE = 0
580      0576 3      END;
581      0577 3
582      0578 3      [%C'-']:
583      0579 4      BEGIN
584      0580 4      | Line continuation indicator
585      0581 4      | Get a new line and reset LIST_PTR
586      0582 4
587      0583 4
588      0584 4      CHSWCHAR_A( 0, CALL_PTR );
589      0585 4      CHSWCHAR_A( 0, ARG_PTR );
590      0586 4
591      0587 4      $XPO GET(IOB=INPUT);
592      0588 4      INPUT_LENGTH = .INPUT[IOBSH_STRING];
593      0589 4      INPUT_PTR = .INPUT[IOBSA_STRING];
594      0590 4      CHSCOPY(.INPUT_LENGTH, .INPUT_PTR,
595      0591 4      | 0,
596      0592 4      | MAX_REC_SIZE, .OUTPUT_PTR);
597      0593 4      | LIST_PTR = .OUTPUT_PTR
598      0594 3      END;
599      0595 3
600      0596 3      [%C':']:
601      0597 3
602      0598 3      | MACRO comment character
603      0599 3      | The remainder of the line is "ignorable"
604      0600 3
605      0601 3      EXITLOOP
606      0602 3      TES
607      0603 2      END;
608      0604 2
609      0605 2
610      0606 2      | Set up implicit outputs as noted
611      0607 2      | If variable number of arguments, then replace CALL_LIST by "%REMAINING"
612      0608 2      | since this is the call argument
613      0609 2
614      0610 2      ARG_LENGTH = CH$DIFF(.ARG_PTR, CH$PTR(ARG_LIST));
615      0611 2      ARG_PTR = CH$PTR(ARG_LIST);
616      0612 2
617      0613 2      IF .VAR_ARGS
618      0614 2      THEN
619      0615 3      BEGIN
620      0616 3      CHSCOPY(
621      0617 3      | CH$LEN_PTR('%REMAINING'),
```

```

: 622 0618 3      0,
: 623 0619 3      MAX_ARG_LIST,
: 624 0620 3      CHSPTR(CALL_LIST));
: 625 0621 3      CALL_LENGTH = %CHARCOUNT('%REMAINING');
: 626 0622 2      END
: 627 0623 2      ELSE
: 628 0624 2      CALL_LENGTH = CH$DIFF(.CALL_PTR, CHSPTR(CALL_LIST));
: 629 0625 2      CALL_PTR = CHSPTR(CALL_LIST)
: 630 0626 2      END;
: 631 0627 1      END;
: INFO#250 L1:0526
: Referenced LOCAL symbol BEGIN_ARG is probably not initialized

```

```

      .PSECT SPLIT$,NOWRT,NOEXE,2
53 47 52 41 5F 4D 55 4E 52 41 56 5F 44 4E 45 00114 P.ABA: .ASCII \END_VARNUM_ARGS\<0>
      00 00 47 4E 49 4E 49 41 4D 45 52 25 00123 P.ABB: .ASCII \%REMAINING\<0>\<0>
      IOBS=           INPUT

```

```
.PSECT $CODE$,NOWRT,2
```

07FC 00000 CONVARGLIST:										
OFFF	8F	00	1008	CA	1010	CA	9E 00014	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10	0447
				6E	00	00	2C 00008	MOVAB	ARG_PTR, R10	0493
					BA	00012	MOVAB	ARG_LIST, ARG_PTR	0494	
OFFF	8F	00			1008	CA	9E 00014	MOVAB	CALL_LIST, CALL_PTR	0495
				6E	00	00	2C 0001B	MOVCS	#0, (SP), #0, #4095, @ARG_PTR	0496
		59	04	AC	1008	DA	00022	ADDL3	#3, TEMP_PTR, LIST_PTR	0497
					2010	CA	D4 0002A	CLRL	VAR_ARGS	0498
						57	D4 0002E 1\$:	CLRL	STATE	0499
						89	9A 00030 2\$:	MOVZBL	(LIST_PTR)+, CHAR	0508
						2D	12 00033	BNEQ	5\$	0512
						01	57 D1 00035	CMPL	STATE, #1	0513
						0C	12 00038	BNEQ	3\$	
						00	BA 3D 90 0003A	MOVB	#61, @ARG_PTR	
						6A	D6 0003E	INCL	ARG_PTR	
						00	BA 30 90 00040	MOVB	#48, @ARG_PTR	
						6A	D6 00044	INCL	ARG_PTR	
						57	D5 00046 3\$:	TSTL	STATE	0522
						15	15 00048	BLEQ	4\$	
OF	50	1008	CA	58	C3 0004A	SUBL3	BEGIN_ARG, CALL_PTR, R0			0526
	00	68	68	50	2D 00050	CMPCS	R0, (BEGIN_ARG), #0, #15, P.ABA			
				0000	CF 00055					
					05 12 00058	BNEQ	4\$			
				2010	CA 01 00 0005A	MOVL	#1, VAR_ARGS			0531
					00CF 31 0005F 4\$:	BRW	17\$			0513
				24	56 D1 00062 5\$:	CMPL	CHAR, #36			0536
					25 13 00065	BEQL	8\$			
				30	56 D1 00067	CMPL	CHAR, #48			

04	AA	6A	50	C3 00135	SUBL3	RO, ARG_PTR, ARG_LENGTH	0611
OFFF	8F	6A	AA	9E 0013A	MOVAB	ARG_LIST, ARG_PTR	0613
		13	08	CA E9 0013E	BLBC	VAR_ARGS 18\$	0620
		CF	2010	OA 2C 00143	MOVCS	#10, P.ABB, #0, #4095, CALL_LIST	Ps
			1010	CA 0014C	MOVL	#10, CALL_LENGTH	--
				OA D0 0014F	BRB	19\$	0621
		100C	CA	OD 11 00154	MOVAB	CALL_LIST, RO	0613
		1008	50	CA 9E 00156 18\$:	SUBL3	RO, CALL_PTR, CALL_LENGTH	0624
		1008	CA	50 C3 0015B	MOVAB	CALL_LIST, CALL_PTR	0626
		1010	CA	9E 00163 19\$:	RET		0627

: Routine Size: 363 bytes, Routine Base: \$CODE\$ + 0521

-L

: 633 0628 1 END
: 634 0629 0 ELUDOM

! End of module MARBLI

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	9292	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	304	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODES	1676	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	116	19	252	00:00.1

Information: 1
Warnings: 0
Errors: 0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:MARBLI/OBJ=OBJ\$:MARBLI MSRC\$:MARBLI/UPDATE=(ENH\$:MARBLI)

Size: 1676 code + 9596 data bytes
Run Time: 00:39.3
Elapsed Time: 01:50.6
Lines/CPU Min: 959
Lexemes/CPU-Min: 74449
Memory Used: 247 pages
Compilation Complete

0233 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

